Paying Due Diligence to Software Architecture in Acquisition

Mike Gagliardi

Tim Morrow

Software Solutions Conference 2015

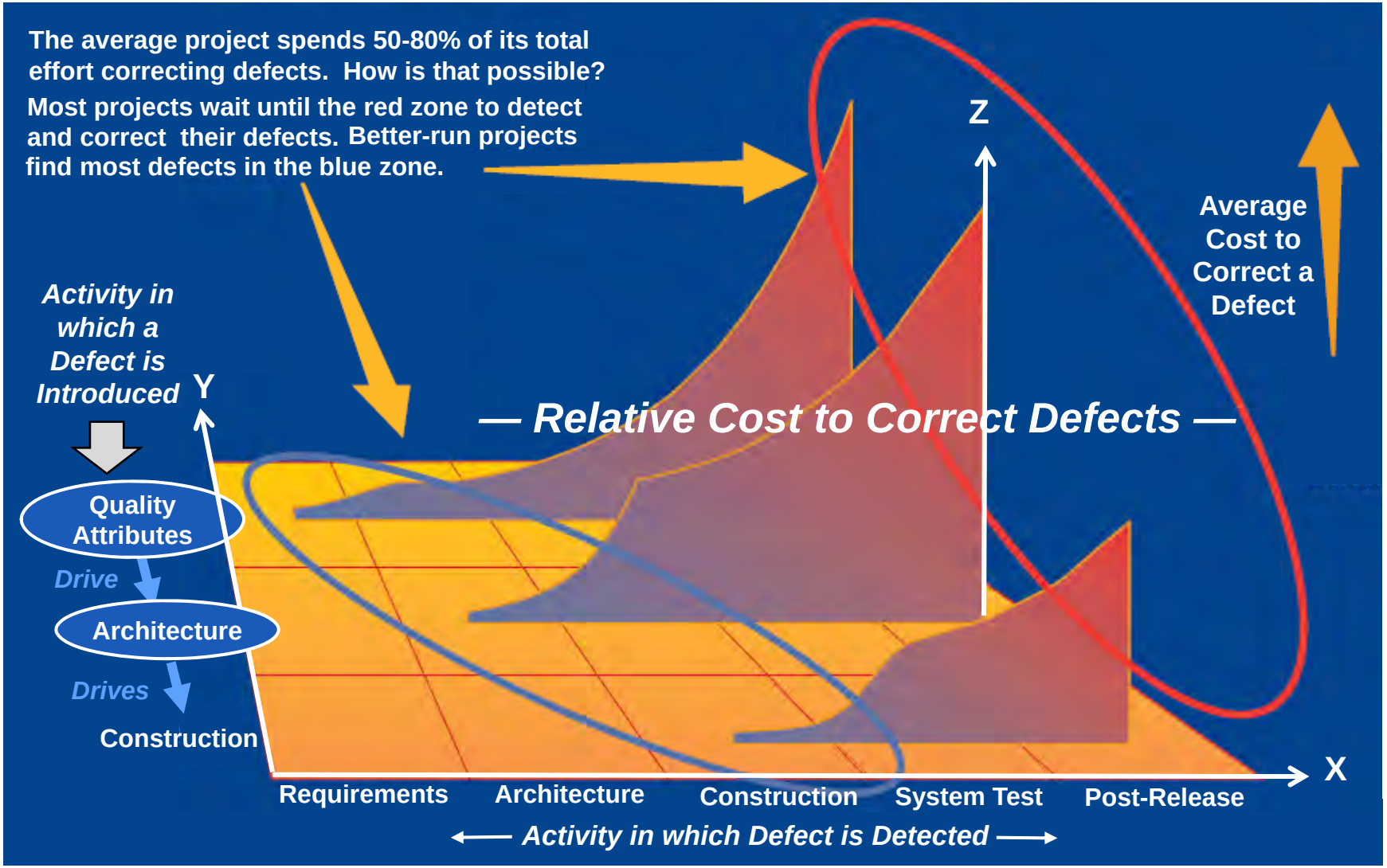
November 16–18, 2015

Software Engineering Institute | Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0002995

Addressing Architecture and Quality Attributes Early in Development Cycle – The Payoff



Source: Adapted from Construx; Software Development Best Practices; *www.construx.com* *copyright © 2008 Construx*

Why Is Architecture So Important?[1]

Architecture is a common high-level communication vehicle for system stakeholders that is amenable to analysis and synthesis.

Architecture embodies the earliest set of design decisions about a system. These decisions

- are the most profound
- are the hardest to get right
- are most difficult to change
- ripple through the entire software development effort
- are most costly to fix downstream
- are critical to achieving mission/business goals

The earlier we reason about architecture tradeoffs, the better.

Why Is Architecture So Important?$_2$

The **right architecture** paves the way for system **success**.

First design artifact that addresses

- **performance**
- **reliability**
- **modifiability**
- **security**

the system's quality attributes

Key to systematic reuse

- **transferable, reusable, analyzable abstraction**

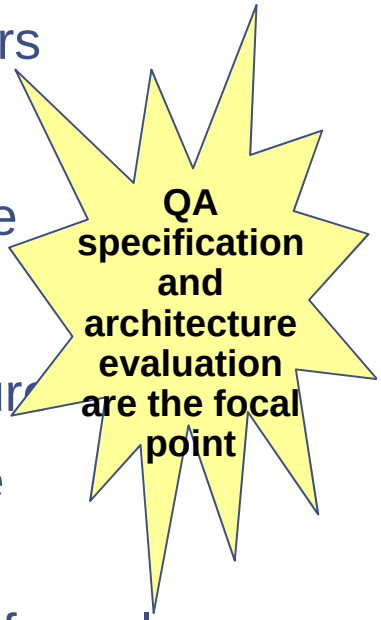
Provides early low-cost means to predict system qualities

- **amenable to analysis and synthesis**
- **amenable to evaluation**

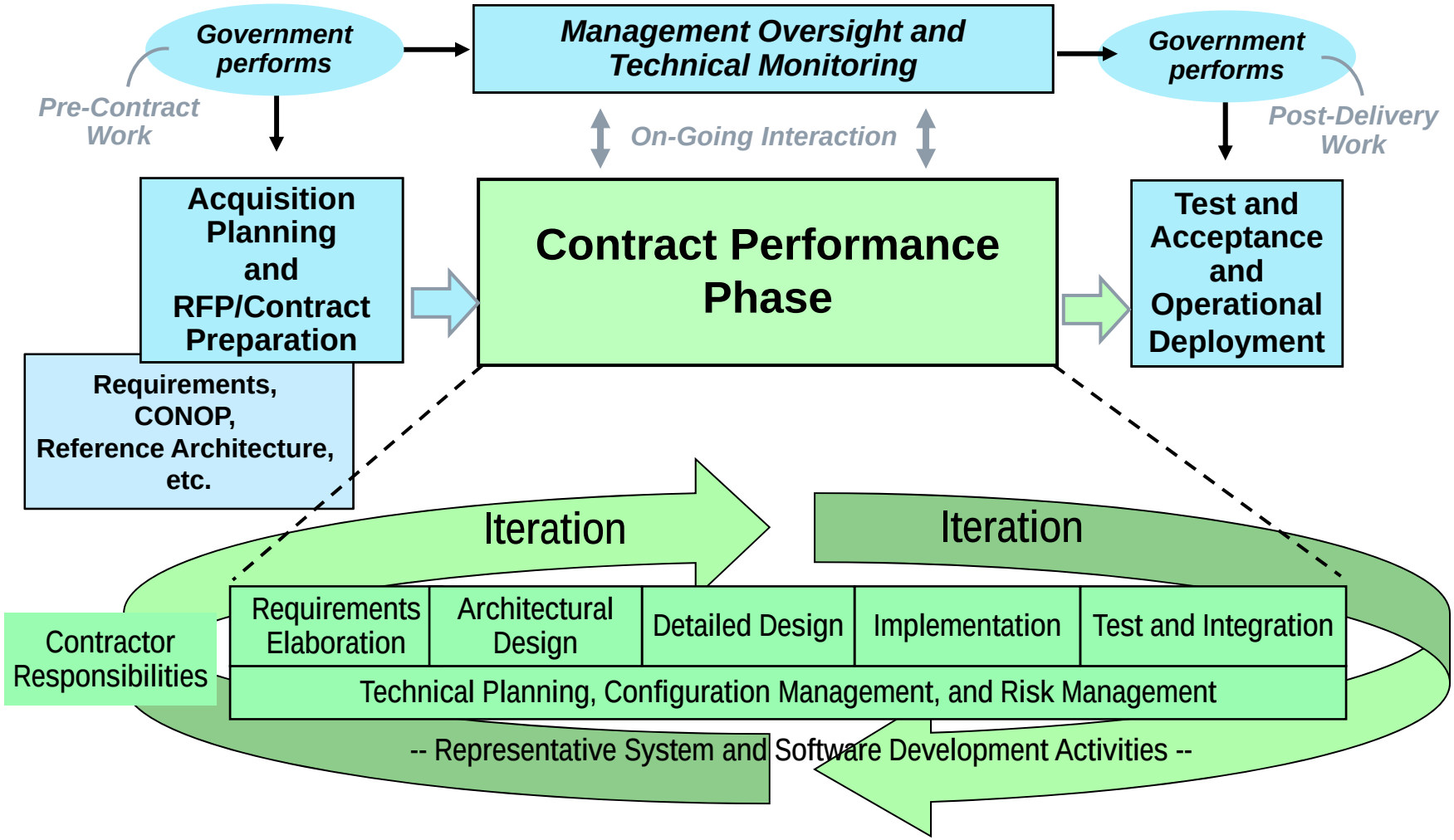
The **wrong architecture** usually spells some form of **disaster**.

Characteristics of an Architecture-Smart Acquisition

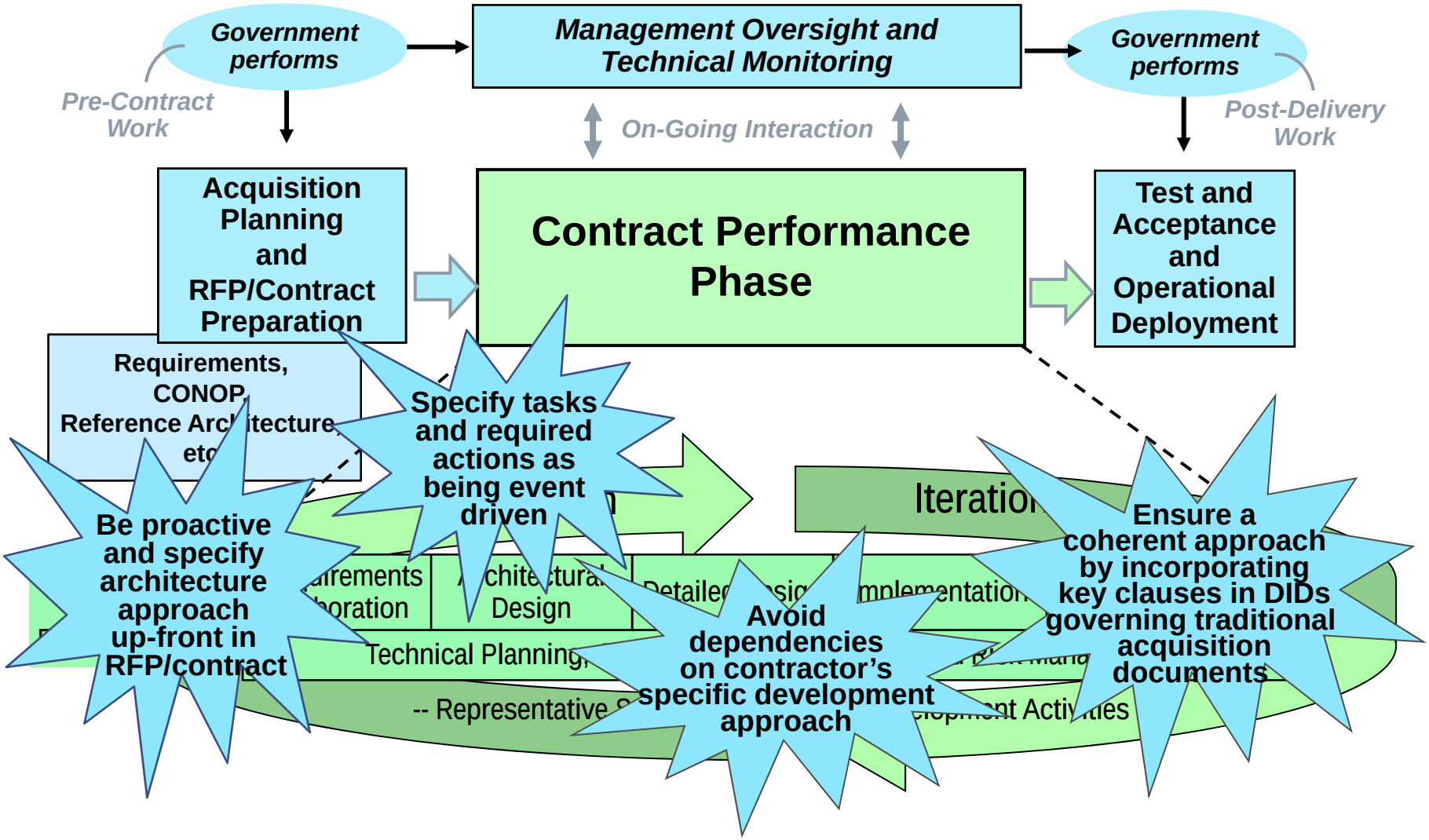
- Understanding the mission drivers for the system being acquired
- Understanding quality attribute expectations of stakeholders
- Developing or selecting the software architecture
- Documenting and communicating the software architecture
- Analyzing and evaluating the software architecture
- Implementing the system based on the software architecture
- Ensuring that the implementation conforms to the software architecture
- Appropriately evolving the architecture over the system's life cycle
- Incorporating other architecture-related management and development activities to achieve specific program objectives



Representation of Contract Performance Phase



Representation of Contract Performance Phase



Typical Scenario Describing Impact of Adopting an Architecture-Smart Acquisition Approach

BEFORE: There is no software architecture documentation.
AFTER: A documented software architecture is a contract deliverable.

BEFORE: The system's non-functional (i.e., quality) requirements that greatly impact the architecture design and software implementation are poorly defined.
AFTER: The system's quality requirements are specified in terms of a clear and concise set of quality attribute scenarios generated by key stakeholders.

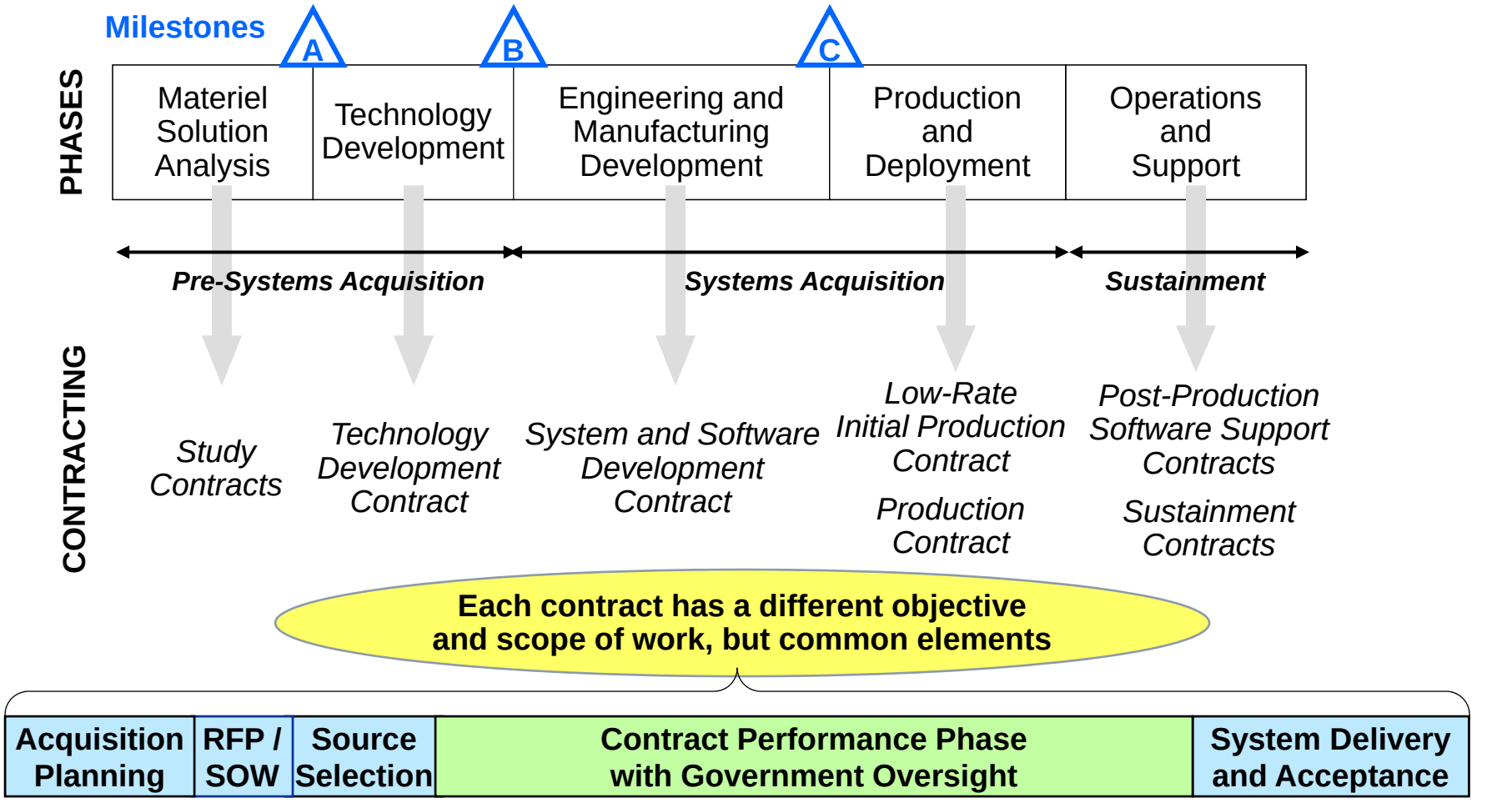
BEFORE: The development contractor presents a couple of PowerPoint box-and-line drawings to describe the architecture and high-level software design.
AFTER: The software architecture description includes a comprehensive set of views (e.g., module decomposition, allocation, run-time) that is amenable to analysis

BEFORE: The proposed software design is not appropriately analyzed or evaluated.
AFTER: The software architecture is evaluated with stakeholder participation and risks (and risk themes) are identified and appropriately documented.

BEFORE: Architecture development is ad hoc and not based on careful analysis.
AFTER: As a result of the architecture evaluation, the development contractor creates a risk mitigation plan and presents it at the Preliminary Design Review (PDR).

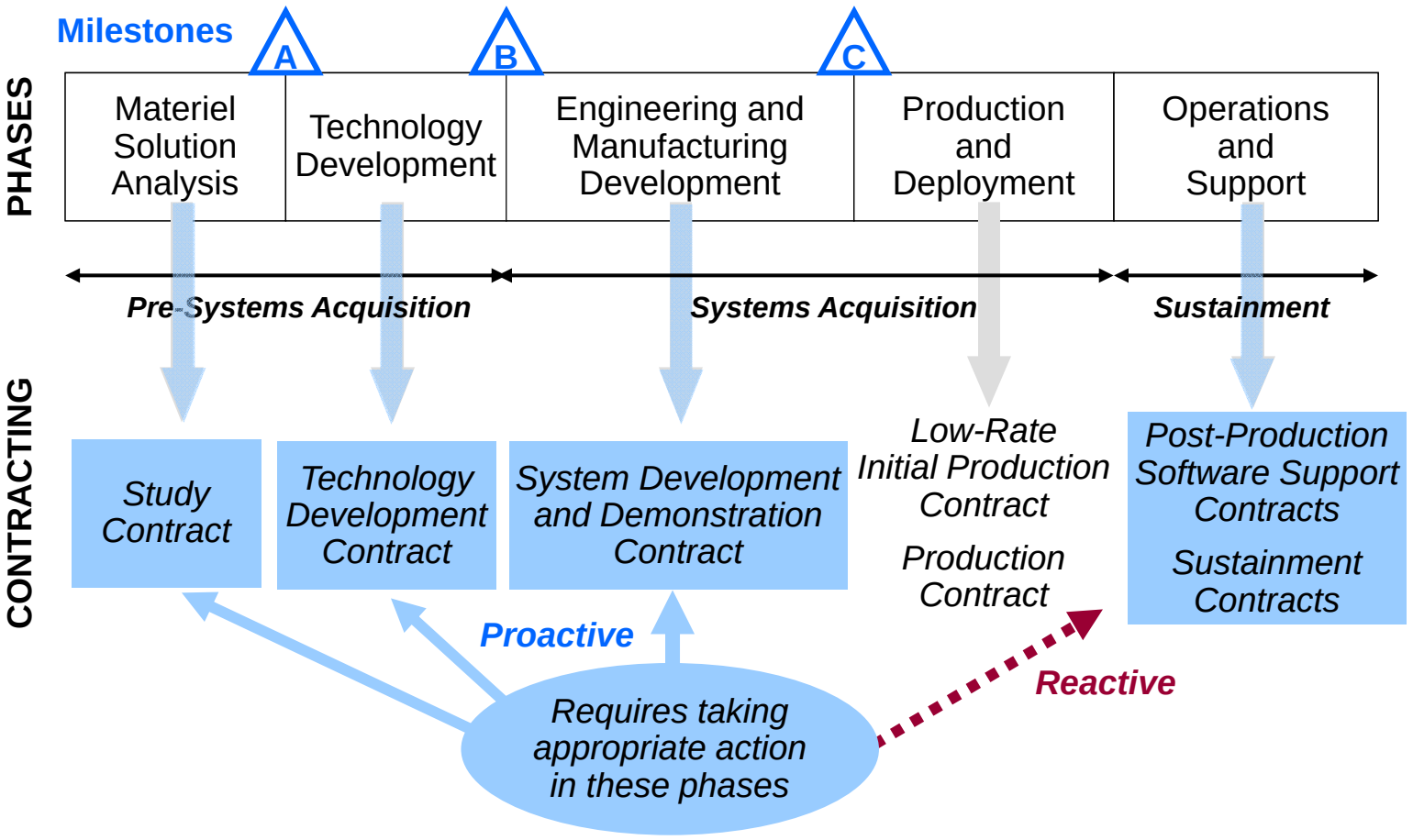
BEFORE: Plans for architecture evolution are ad hoc and not based on careful analysis.
AFTER: In conjunction with the risk mitigation plan the development contractor develops a software architecture improvement roadmap based on an incremental software development approach.

Contractual View of DoD Acquisition Life Cycle



Contractual View of DoD Acquisition Life Cycle

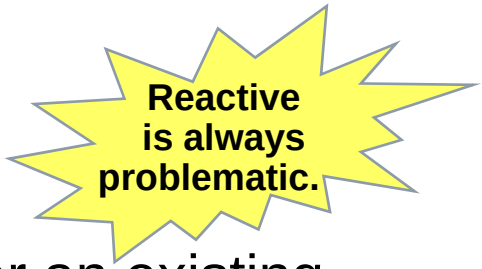
– Adopting an Architecture-Smart Acquisition Approach –



Two Fundamental Ways Architecture-Smart Activities can be Incorporated in an Acquisition

Reactive

Software Architecture activities are *initiated opportunistically* and performed in situ under an existing contract at the request of the program manager.[1]

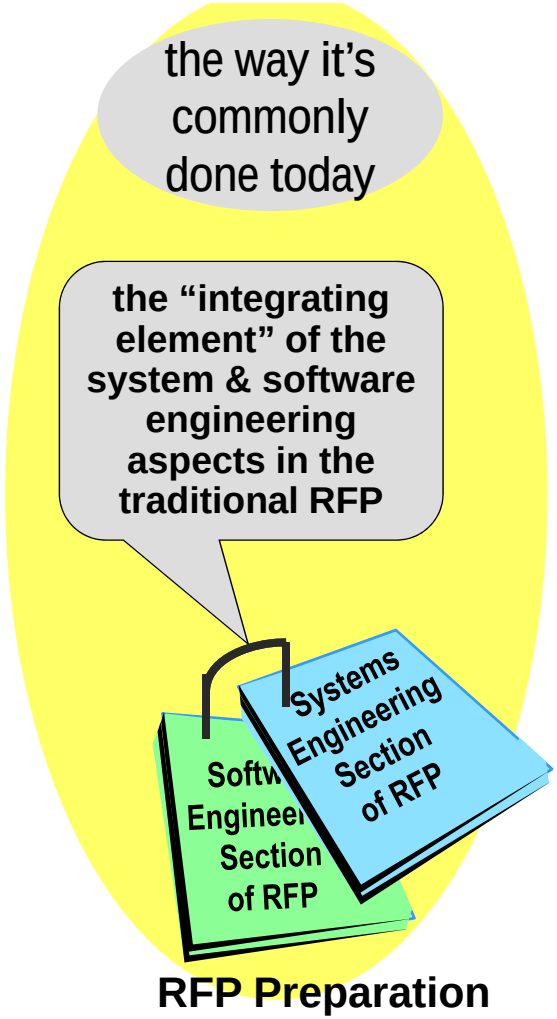


Proactive

Software Architecture activities are *preplanned and integrated* up front in a request for proposal (RFP) for a system (or software) acquisition.

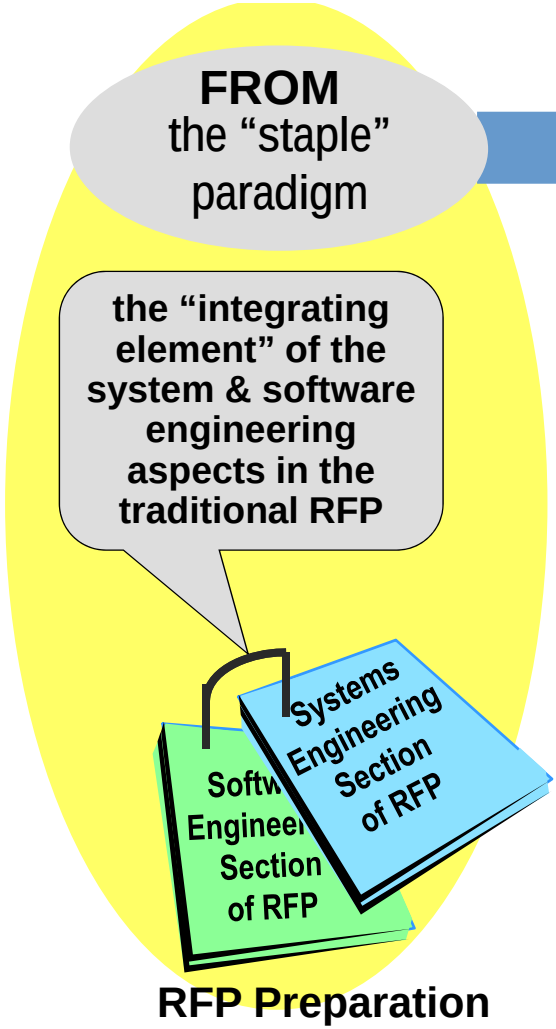
[1] Or at the request of a contractor under a negotiated agreement

Integration of Systems and Software Engineering Aspects in an RFP



RFP Preparation

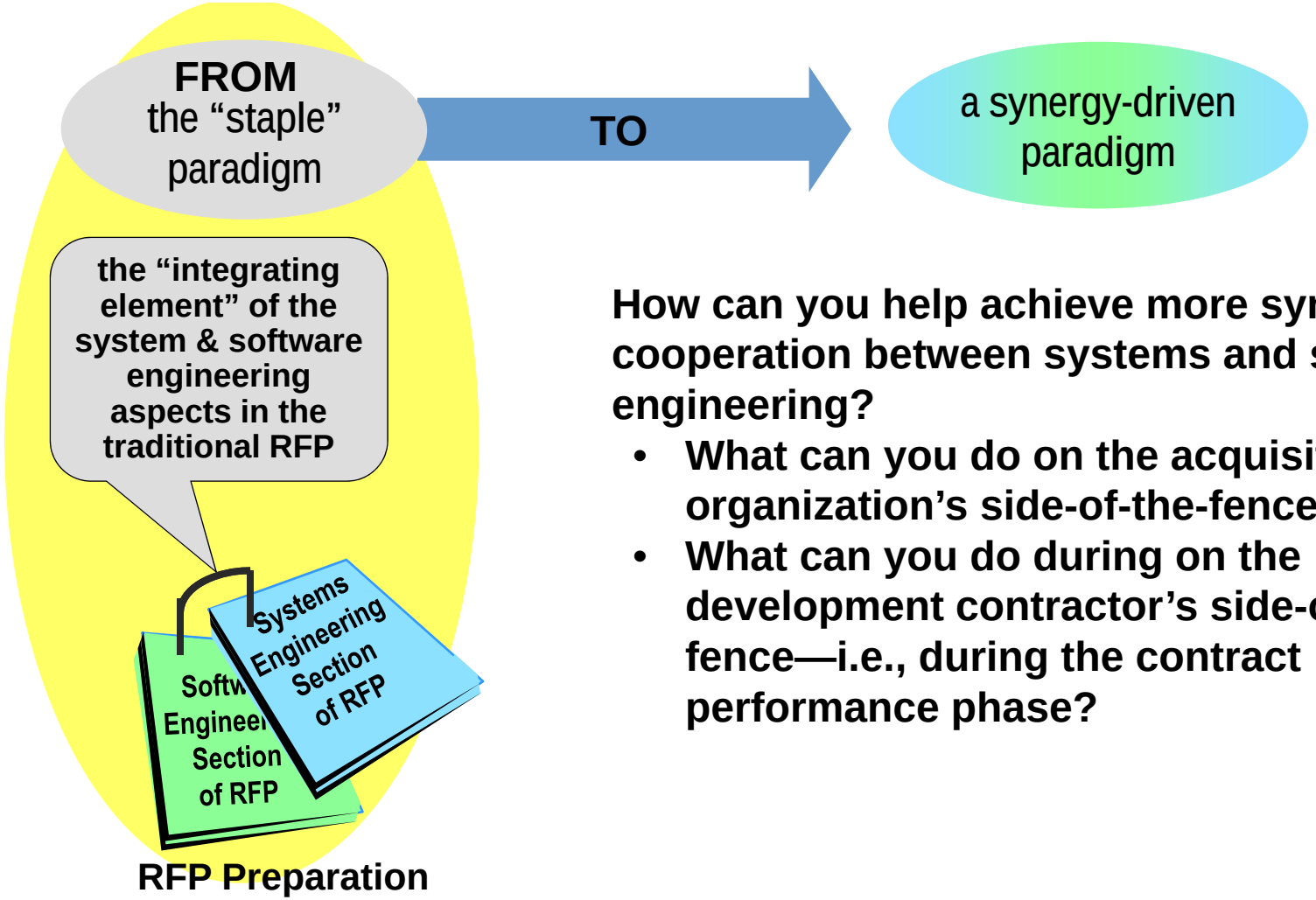
Promoting System and Software Engineering Congruency in Acquisition



Definition of Synergy

1. The interaction of two or more agents or forces so that their combined effect is greater than the sum of their individual effects.
2. Cooperative interaction among groups, especially among the acquired subsidiaries or merged parts of a corporation, that creates an enhanced combined effect.

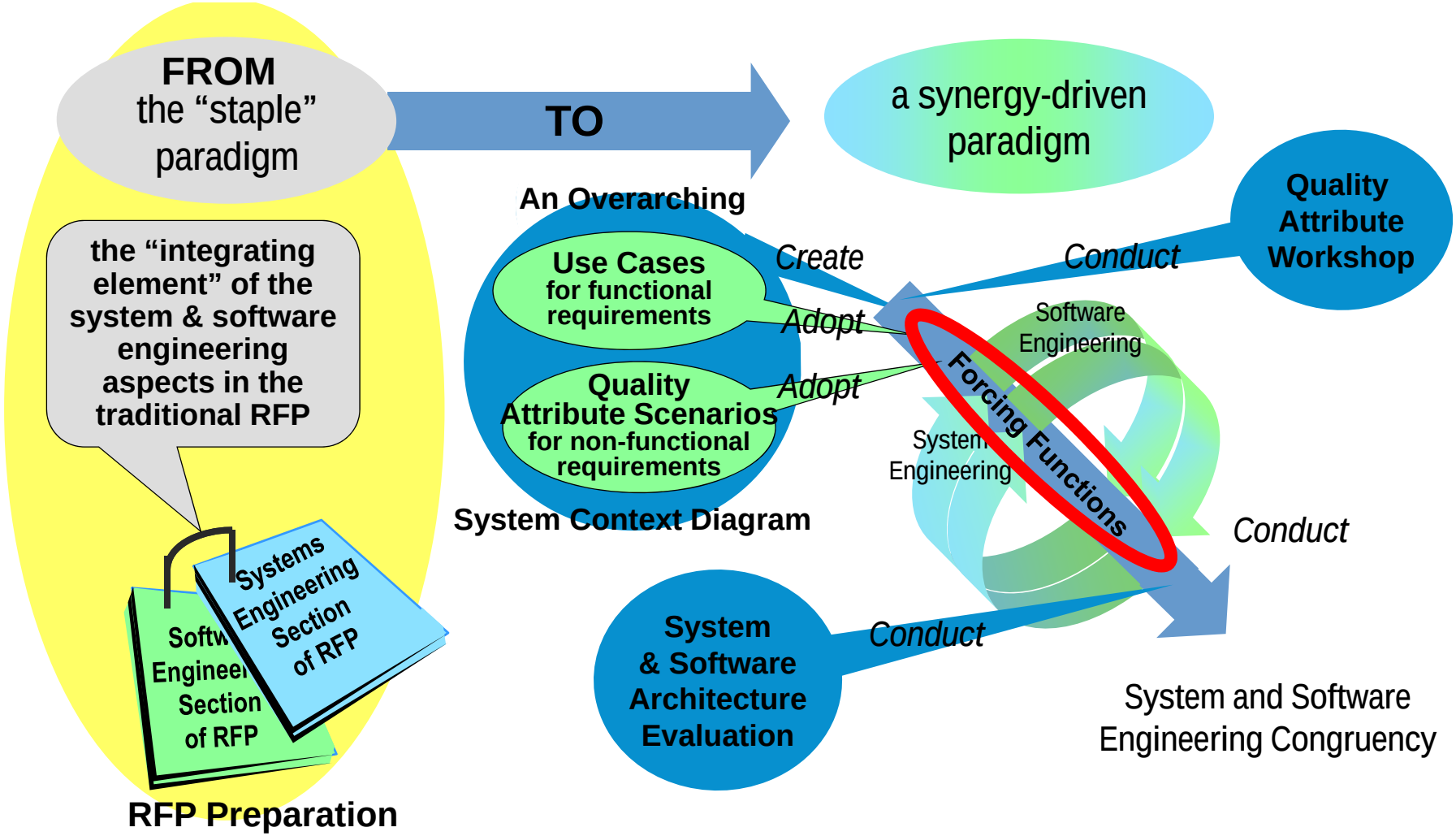
Promoting System and Software Engineering Congruency in Acquisition



How can you help achieve more synergy and cooperation between systems and software engineering?

- **What can you do on the acquisition organization's side-of-the-fence?**
- **What can you do during on the development contractor's side-of-the fence—i.e., during the contract performance phase?**

Promoting System and Software Engineering Congruency in Acquisition



Key Elements of an Architecture-Smart Acquisition and Development Approach

1. Specifying a system's quality attributes

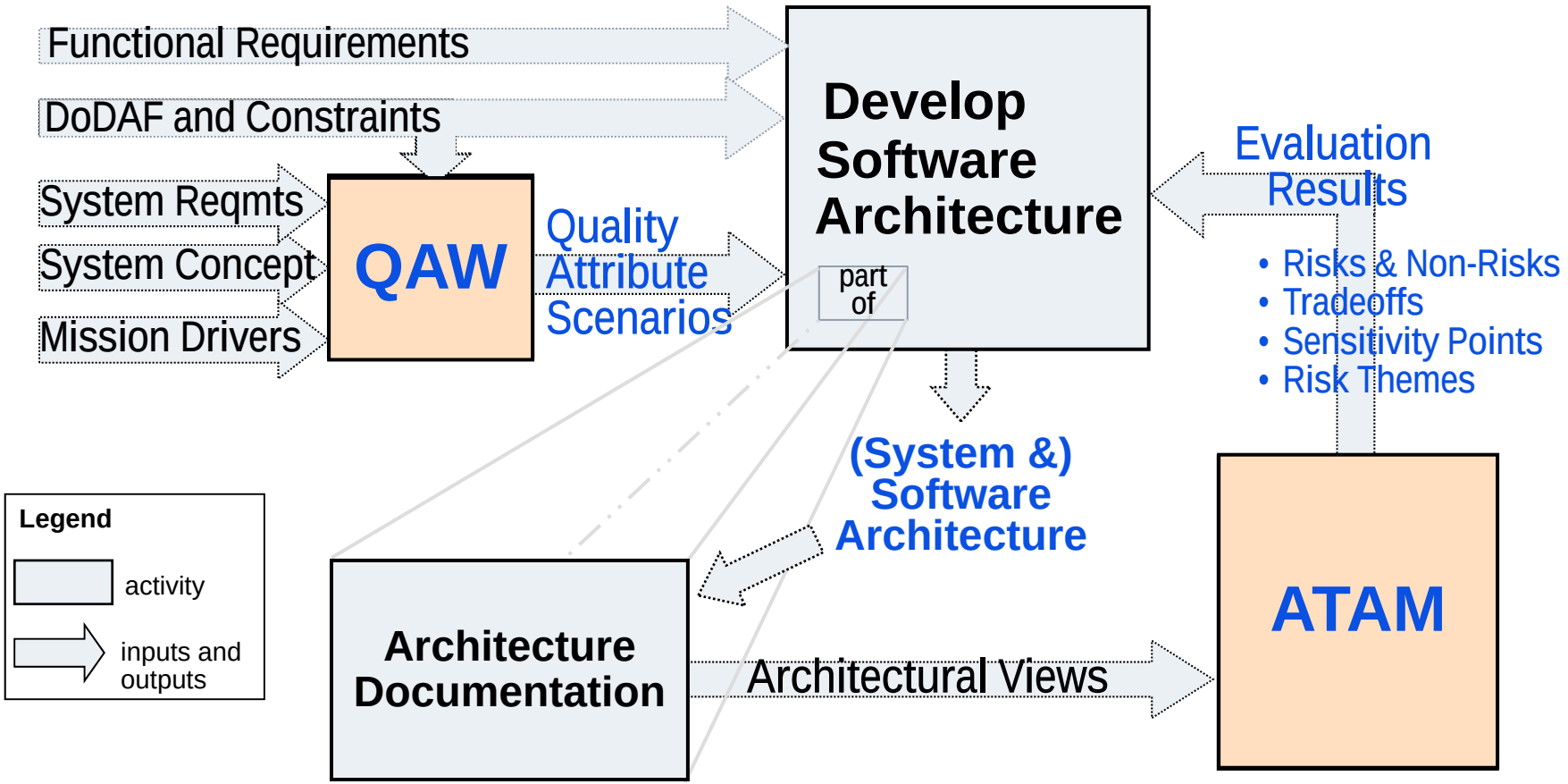
This involves conducting a Quality Attribute Workshop (QAW) with key stakeholders to elicit and capture[1] quality attribute scenarios (i.e., specify the non-functional requirements) so the architecture can be appropriately designed.

2. Evaluating the system and software architecture

This involves conducting an architecture evaluation (in collaboration with the system developer) using the system variant of the SEI's Architecture Tradeoff and Analysis Method to identify and mitigate risks early in the system development cycle.

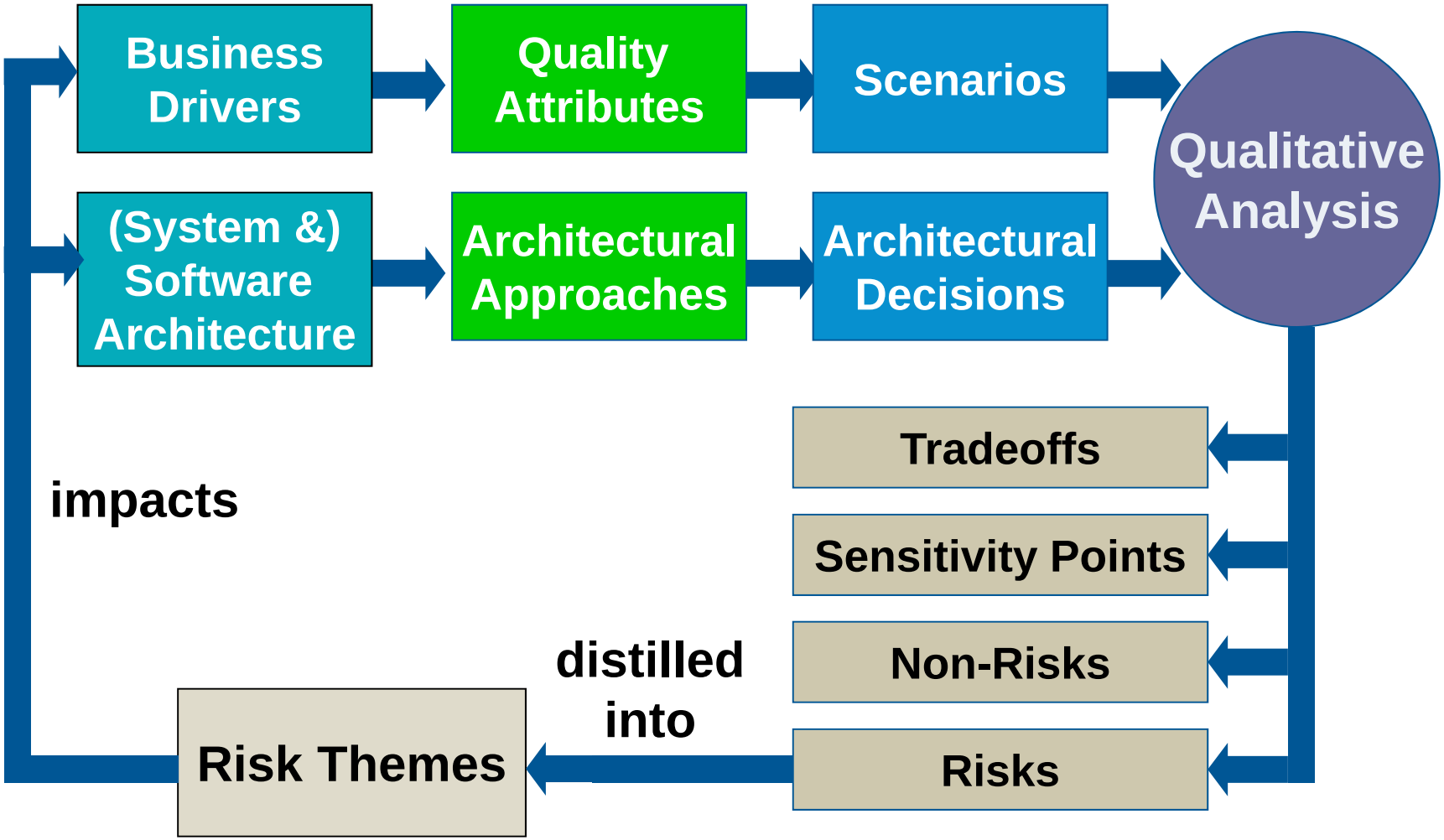
[1] To represent and record in a lasting form

Conceptual Flow

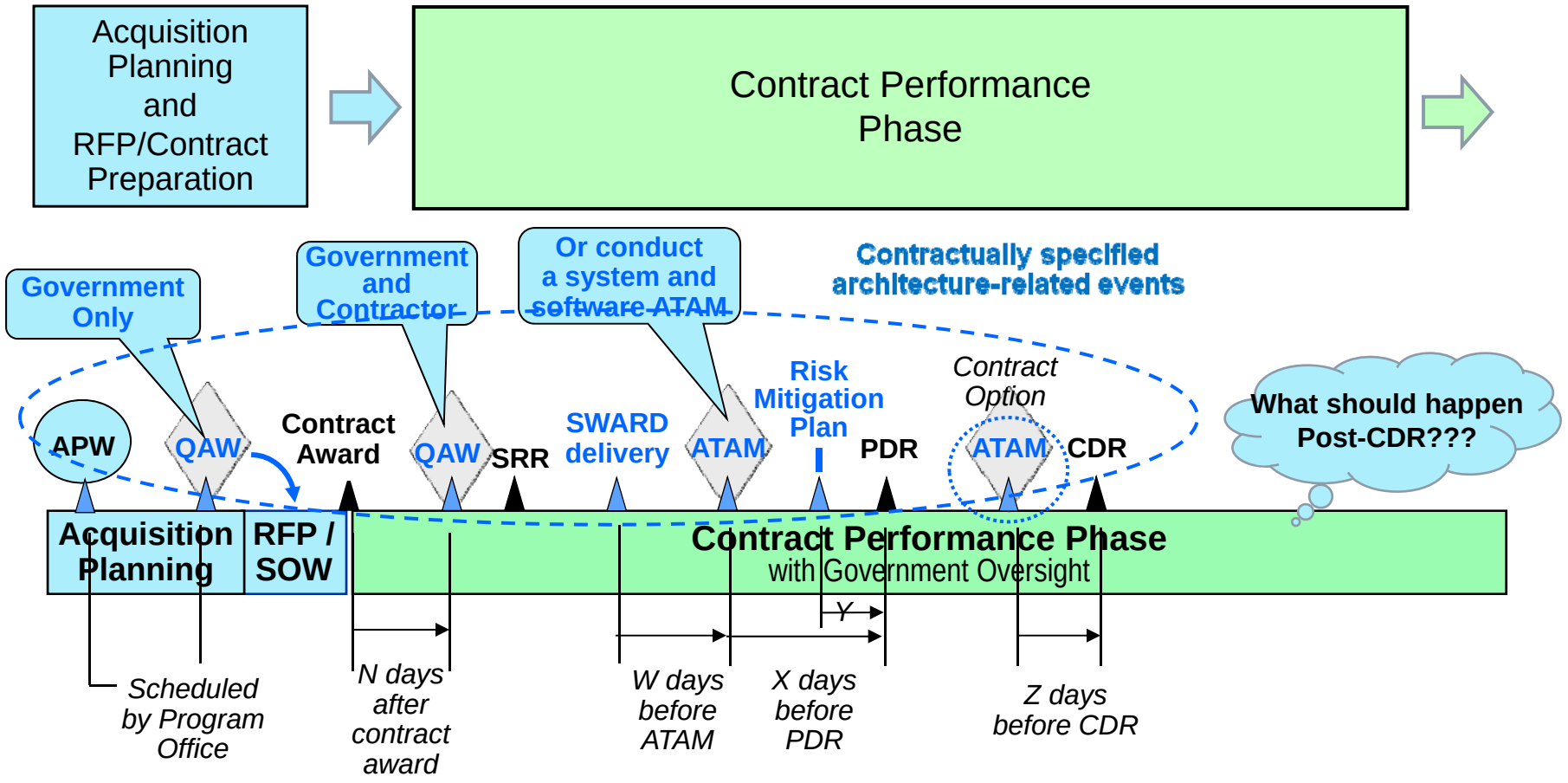


Such a "big picture" view of a contractor's architecture development approach would be described in its Software Development Plan (SDP).

Conceptual Flow of ATAM



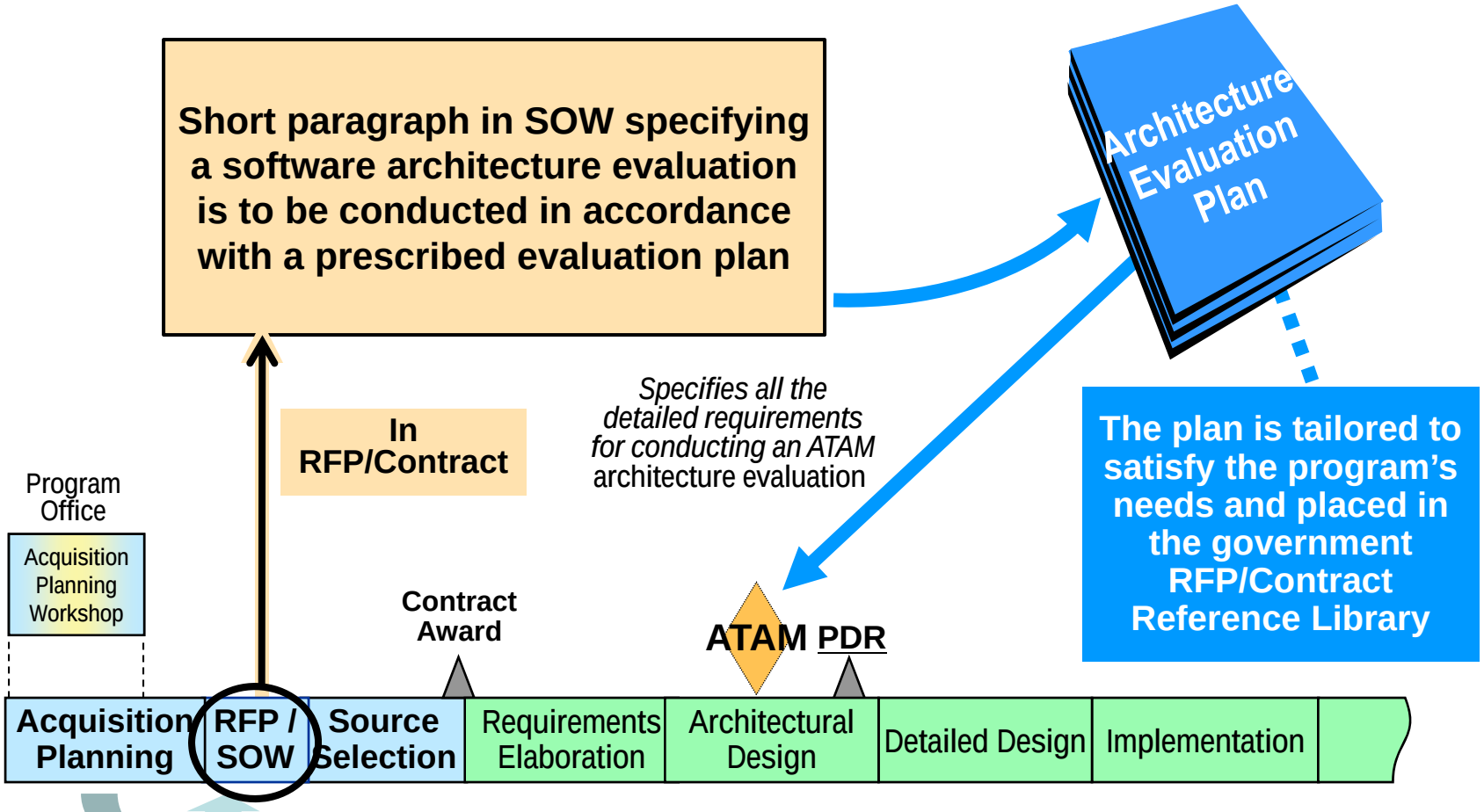
Elements of an Architecture-Smart Acquisition



Legend

APW – Acquisition Planning Workshop
PDR – Preliminary Design Review
QAW – Quality Attribute Workshop
SRR – System Requirements Review
SWARD – Software Architecture Description Document
CDR – Critical Design Review
ATAM – Architecture Trade-off Analysis Method

Model for Incorporating Software Architecture Evaluation in an RFP/Contract



* See http://www.sei.cmu.edu/library/abstracts/reports/09tn004.cfm for an example description of an ATAM plan

Instantiation → Elements of an Architecture-Smart Acquisition Approach

Ensuring a Coherent Approach is Adopted

Document	Type of Information to Be Included (Relative to Conducting an Architecture Evaluation)
SEMP	Describe: (1) how the architecture evaluation is integrated into the system engineering management plan in relation to the program milestones, (2) how the system's quality attribute requirements (i.e., nonfunctional requirements) that drive the architectural design will be specified and managed, and (3) how the software architecture will be documented.
TEMP	Describe the role of architecture evaluation in the test and evaluation management plan and when the evaluation will be scheduled in relation to the program milestones.
SEP	Describe: (1) how the architecture evaluation is integrated into the system engineering plan in relation to the system engineering milestones, (2) how the system's quality attribute requirements (i.e., nonfunctional requirements) that drive the architectural design will be specified and managed, and (3) how the software architecture will be documented.
SDP	Describe how the software architecture evaluation fits into the overall software development approach including how identified risks (and risk themes) will be analyzed and mitigated.
STP	Describe the role of architecture evaluation in the software test plan and when the evaluation will be scheduled in relation to software testing milestones.
RMP	Describe how risks (and risk themes) emanating from the architecture evaluation will be integrated with the program's risk management system and subsequently managed (i.e., identified, tracked, and mitigated).

Example: Architecture Aspects You May Want the Offeror to Describe in their Technical Proposal

Section L – Instructions to Offerors

1. Describe how quality attribute scenarios resulting from the QAW will be integrated into the requirements baseline and managed from that point forward.
2. Describe how architecture risks and risk themes discovered during the ATAM evaluation will be prioritized and mitigated.
3. Describe how proposed software modifications (including architectural changes) that occur during the system life cycle will be managed.
4. Describe how compliance of the software implementation with the approved software architecture baseline will be enforced throughout the life cycle.
5. Describe what kind of software architecture metrics will be collected and reported to the government during the contract performance phase.

Examples of Architecture Practices that Can Be Incorporated in an RFP/Contract

Architecture activities, deliverables and measures that can easily be incorporated into an RFP today for a system acquisition include:

- **Quality Attribute Workshop (QAW)** to collaboratively
  - Validate business and mission drivers
  - Elicit quality attributes scenarios (system and software)
  - Refine a set of quality attribute scenarios
- **Architecture design and evolution guidance**
  - Quality attribute-driven architectural design
- **Software architecture description**
  - Include as part of contractual deliverables
- **(System &) Software Architecture Evaluation**
  - Specify collaborative evaluation based on ATAM
  - Require evaluation report identifying risks and risk themes
- **Risk mitigation "monitoring instrument"**
  - Monitor the risk mitigation activities and report on progress
- **Cost benefit change analysis**
  - Prioritization of architecture risk mitigation activities based on cost benefit

Conducting an Acquisition Planning Workshop

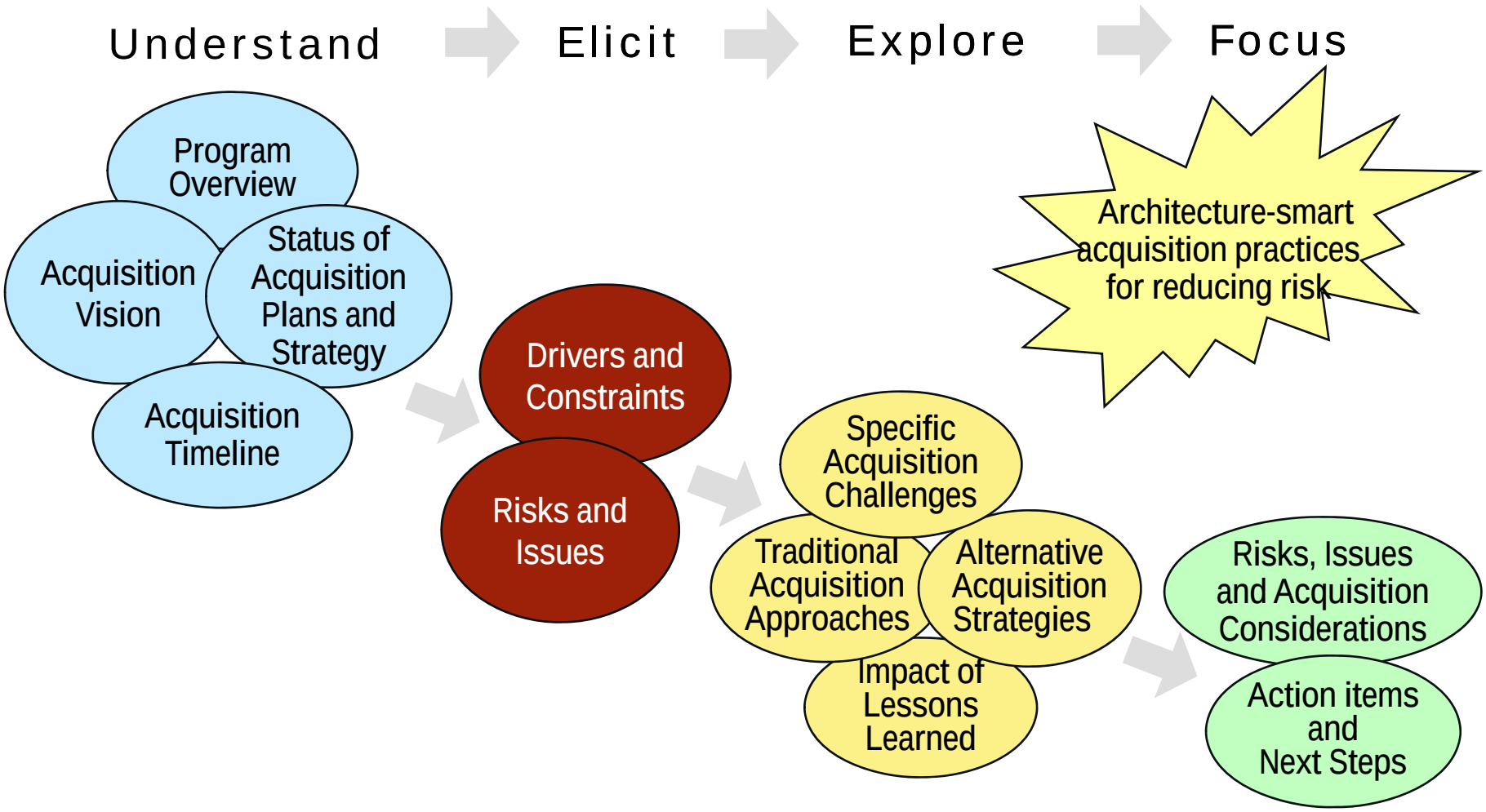
Why hold a workshop?

1. To be **proactive** and provide upfront assistance during the **acquisition planning and RFP preparation phase** when it can make a difference.
2. To provide a **structured forum** for key acquisition stakeholders to **understand** the program's acquisition approach and current status, **and explore** potential ways for **reducing software acquisition risk** via a facilitated technical interchange

Outputs

1. Common understanding of the acquisition challenges, risks, and key issues
2. A list of actions for going forward with acquisition planning

Overview of Acquisition Planning Workshop



How Acquisition Programs Can Leverage an Architecture-Smart Approach to Reduce Risk

Realize that Architecture is Key

- Embodies the early design decisions that are the most difficult to get right
- Provides level-of-abstraction best aligned with program responsibilities

Focus on Quality Attributes

- Allows stakeholders to discuss, clarify, and prioritize non-functional requirements that are often problematic

Acquire Architecture Documentation

- Provides the means to analyze the software design and guide development

Evaluate the System and Software Architecture

- Promotes coordination between system and software engineering

Focus on Risk Management

- Risk identification and mitigation

Arrange for Training

- Educate both program office and contract personnel

Conduct an Acquisition Planning Workshop

- Be proactive and endure the right stuff gets in the RFP/contract

Contact Information

Mike Gagliardi

Principal Engineer

Software Engineering Institute

Office: 412-268-7738

Email: mjg@sei.cmu.edu